Amendments to the Claims:

A listing of the entire set of pending claims (including amendments to the claims, if any) is submitted herewith per 37 CFR 1.121. This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently amended) A method of generating a common secret between a first ~~party~~ machine and a second ~~party~~ machine, ~~in which~~ wherein the first ~~party~~ machine holds a value p_1 and a symmetrical polynomial $P(x,y)$ fixed in the first argument by the value p_1, and the first ~~party~~ machine performs the steps of:

sending the value p_1 to the second ~~party~~ machine,

receiving a value p_2 from the second ~~party~~ machine and

calculating the common secret S_1 by evaluating the polynomial $P(p_1, y)$ in p_2,

wherein the first ~~party~~ machine additionally holds a value q_1 and a symmetrical polynomial $Q(x, z)$ fixed in the first argument by the value q_1, and further performs the steps of:

sending q_1 to the second ~~party~~ machine,

receiving a value q_2 from the second ~~party~~ machine and

calculating the secret S_1 as $S_1=Q(q_1, q_2)\cdot P(p_1, p_2)$.

2. (Currently amended) The method of claim 1, ~~in which~~ wherein the first ~~party~~ machine further performs the steps of:

obtaining a random number r_1,

calculating $r_1\cdot q_1$, sending $r_1\cdot q_1$ to the second ~~party~~ machine,

receiving $r_2\cdot q_2$ from the second ~~party~~ machine, and

calculating the secret S_1 as $S_1=Q(q_1, r_1\cdot r_2\cdot q_2)\cdot P(p_1, p_2)$.

3. (Currently amended) The method of claim 2, ~~in which~~ wherein the first ~~party~~ machine holds the value q_1 multiplied by an arbitrarily chosen value r, and the product $Q(q_1, z)P(p_1, y)$ instead of the individual polynomials $P(p_1, y)$ and $Q(q_1, z)$, and the first ~~party~~ machine performs the steps of:

______calculating $r_1 \cdot r \cdot q_1$,

______sending $r_1 \cdot r \cdot q_1$ to the second ~~party~~ machine,

______receiving $r_2 \cdot r \cdot q_2$ from the second ~~party~~ machine, and

______calculating the secret S_1 as $S_1 = Q(q_1, r_1 \cdot r_2 \cdot r \cdot q_2) \cdot P(p_1, p_2)$.

4. (Currently amended) The method of claim 1, ~~in which~~ wherein the second ~~party~~ machine holds a value p_2 and a value q_2, the symmetrical polynomial $P(x, y)$ fixed in the first argument by the value p_2, the symmetrical polynomial $Q(x, z)$ fixed in the first argument by the value q_2, and the second ~~party~~ machine performs the steps of:

______sending q_2 to the first ~~party~~ machine,

______receiving q_1 from the first ~~party~~ machine and

______calculating a secret S_2 as $S_2 = Q(q_2, q_1) \cdot P(p_2, p_1)$, ~~whereby~~ the common secret ~~has been~~ being generated if the secret S_2 equals the secret S_1.

5. (Currently amended) The method of claim 1, ~~in which~~ wherein a trusted third ~~party~~ machine performs the steps of:

______choosing a symmetric $(n+1) \times (n+1)$ matrix T,

______constructing the polynomial P using entries from the matrix T as respective coefficients of the polynomial P,

______constructing the polynomial $Q(x, y)$,

______choosing the value p_1, the value p_2, the value q_1 and the value q_2,

______sending the value p_1, the value q_1, the polynomial $P(x, y)$ fixed in the first argument by the value p_1 and the polynomial $Q(x, z)$ fixed in the first argument by the value q_1 to the first ~~party~~ machine, and

______sending the value p_2, the value q_2, the polynomial $P(x, y)$ fixed in the first argument by the value p_2 and the polynomial $Q(x, z)$ fixed in the first argument by the value q_2 to the second ~~party~~ machine

6. (Currently amended) The method of claim 5, ~~in which~~ wherein the trusted third ~~party~~ machine further:

arbitrarily chooses a value r,

sends the value $r \cdot q_1$ instead of the value q_1 and the product $Q(q_1, z)P(p_1, y)$ instead of the individual polynomials $P(p_1, y)$ and $Q(q_1, z)$ to the first ~~party~~ machine and

sends the value $r \cdot q_2$ instead of the value q_2 and the product $Q(q_2, z)P(p_2, y)$ instead of the individual polynomials $P(p_2, y)$ and $Q(q_2, z)$ to the second ~~party~~ machine.

7. (Currently amended) The method of claim 5, ~~in which~~ wherein the trusted third ~~party~~ machine further performs the steps of

choosing a set comprising *m* values p_i, including the values p_1 and p_2,

calculating a space **A** from the tensor products $\vec{p}_i^V \otimes \vec{p}_j^V$ of the Vandermonde vectors $\vec{p}_i^V$ built from the set of values p_i,

choosing a vector $\vec{\gamma}_1$ and a vector $\vec{\gamma}_2$ from the perpendicular space $\mathbf{A}^\perp$ of the space **A**,

constructing a matrix $T_{\Gamma_1} = T+\Gamma_1$ from the vector $\vec{\gamma}_1$ and a matrix $T_{\Gamma_2} = T+\Gamma_2$ from the vector $\vec{\gamma}_2$,

constructing a polynomial $P^{\Gamma_1}(x, y)$ using entries from the matrix T_{Γ_1} and

sending the polynomial $P^{\Gamma_1}(x, y)$ fixed in the first argument by the value p_1 to the first ~~party~~ machine, and

constructing a polynomial $P^{\Gamma_2}(x, y)$ using entries from the matrix T_{Γ_2} and

sending the polynomial $P^{\Gamma_2}(x, y)$ fixed in the first argument by the value p_2 to the second ~~party~~ machine.

8. (Currently amended) The method of claim 5, ~~in which~~ wherein a number *m'* of values p_i, and $m' < m$, are distributed to additional parties.

9. (Currently amended) The method of claim 1, ~~in which~~ wherein the first ~~party~~ machine and the second ~~party~~ machine use a non-linear function on the generated secret S1 and S2, respectively, before using it as a secret key in further communications.

10. (Currently amended) The method of claim 9, wherein ~~in which~~ a one-way hash function is applied to the generated secrets S1 and S2.

11. (Currently amended) The method of claim 9, wherein ~~in which~~ a non-linear function in the form of a polynomial is applied to the generated secrets S1 and S2.

12. (Currently amended) The method of claim 1, further comprising the step of verifying that the second ~~party~~ machine knows the secret S_1.

13. (Currently amended) The method of claim 12, ~~in which~~ wherein the first ~~party~~ machine subsequently applies a zero-knowledge protocol to verify that the second ~~party~~ machine knows the secret S_1.

14. (Currently amended) The method of claim 12, ~~in which~~ wherein the first ~~party~~ machine subsequently applies a commitment-based protocol to verify that the second ~~party~~ machine knows the secret S_1.

15. (Currently amended) The method of claim 14, ~~in which~~ wherein the second ~~party~~ machine uses a symmetric cipher to encrypt a random challenge, and sends the encrypted random challenge to the first ~~party~~ machine and the first ~~party~~ machine subsequently uses the same symmetric cipher as a commit function to commit ~~himself~~ itself to a decryption of the encrypted random challenge.

16. (Currently amended) A system comprising:

a first ~~party~~ machine,

a second ~~party~~ machine and

a trusted third ~~party~~ machine, that is arranged to generate a common secret between the first ~~party~~ machine and the second ~~party~~ machine, ~~in which~~

wherein:

the first ~~party~~ machine holds a value p_1 and a symmetrical polynomial P(x,y) fixed in the first argument by the value p_1, and the first ~~party~~ machine performs the steps of: sending the value p_1 to the second ~~party~~ machine,

receiving a value p_2 from the second ~~party~~ machine and

calculating the common secret S_1 by evaluating the polynomial $P(p_1, y)$ in p_2, ~~wherein~~ and

the first ~~party~~ machine additionally holds a value q_1 and a symmetrical polynomial Q(x, z) fixed in the first argument by the value q_1, and further performs the steps of:

sending q_1 to the second ~~party~~ machine,

receiving a value q_2 from the second ~~party~~ machine and

calculating the secret S_1 as $S_1=Q(q_1, q_2)\cdot P(p_1, p_2)$.

17. (Currently amended) A device (P) ~~arranged to~~ comprising:

a memory that is configured to hold a value p_1, a symmetrical polynomial P(x,y) fixed in the first argument by the value p_1, a value q_1, and a symmetrical polynomial Q(x, z) fixed in the first argument by the value q_1,

a transceiver that is configured to send the value p_1 to a second ~~party~~ machine, and receive a value p_2 from the second ~~party~~ machine, and

a processor that is configured to evaluate the polynomial $P(p_1, y)$ in p_2,

wherein:

the transceiver is configured to send q_1 to the second ~~party~~ machine, ~~receiving~~ and receive a value q_2 from the second ~~party~~ machine, and

the processor is configured to evaluate the polynomial $Q(q_1, q_2)$, and calculate a secret S_1 as $S_1=Q(q_1, q_2)\cdot P(p_1, p_2)$.

18. (Currently amended) The device of claim 17, ~~comprising storage means for storing~~ wherein the memory is configured to store the polynomial P and the polynomial Q in the form of their respective coefficients.

19. (Currently amended) A computer readable media that includes a program product for causing one or more processors to generate a common secret between a first ~~party~~ machine and a second ~~party~~ machine, ~~in which~~ wherein the first ~~party~~ machine holds a value p_1 and a symmetrical polynomial P(x,y) fixed in the first argument by the value p_1, and the first ~~party~~ machine performs the steps of:

sending the value p_1 to the second ~~party~~ machine,

receiving a value p_2 from the second ~~party~~ machine and

calculating the common secret S_1 by evaluating the polynomial $P(p_1, y)$ in p_2,

wherein the first ~~party~~ machine additionally holds a value q_1 and a symmetrical polynomial Q(x, z) fixed in the first argument by the value q_1, and further performs the steps of:

sending q_1 to the second ~~party~~ machine,

receiving a value q_2 from the second ~~party~~ machine and

calculating the secret S_1 as $S_1=Q(q_1, q_2)\cdot P(p_1, p_2)$.

20. (Currently amended) The system of claim 16, wherein the second ~~party~~ machine holds a value p_2 and a value q_2, the symmetrical polynomial P(x, y) fixed in the first argument by the value p_2, the symmetrical polynomial Q(x, z) fixed in the first argument by the value q_2, and the second ~~party~~ machine performs the steps of:

sending q_2 to the first ~~party~~ machine,

receiving q_1 from the first ~~party~~ machine and

calculating a secret S_2 as $S_2=Q(q_2, q_1)\cdot P(p_2, p_1)$, ~~whereby~~ the common secret ~~has been~~ being generated if the secret S_2 equals the secret S_1.